



\*\*FILE\*\*ID\*\*REMOVE

E 7

RRRRRRRR R EEEEEEEEEE MM MM 000000 VV VV EEEEEEEEEE  
RRRRRRRR RR EEEEEE MM MM 000000 VV VV VV VV EEEEEE  
RR RR EE MMMM MMMM 00 00 VV VV VV VV EE  
RR RR EE MMMM MMMM 00 00 VV VV VV VV EE  
RR RR EE MM MM 00 00 VV VV VV VV EE  
RR RR EE MM MM 00 00 VV VV VV VV EE  
RRRRRRRR EEEEEEEE MM MM 00 00 VV VV VV VV EEEEEE  
RRRRRRRR EEEEEEEE MM MM 00 00 VV VV VV VV EEEEEE  
RR RR EE MM MM 00 00 VV VV VV VV EE  
RR RR EE MM MM 00 00 VV VV VV VV EE  
RR RR EEEEEEEEEE MM MM 000000 VV VV EEEEEEEEEE  
RR RR EEEEEEEEEE MM MM 000000 VV VV EEEEEEEEEE

....  
....  
....  
....

CCCCCCCC 000000 MM MM  
CCCCCCCC 000000 MM MM  
CC 00 00 MMMM MMMM  
CC 00 00 MMMM MMMM  
CC 00 00 MM MM MM  
CCCCCCCC 000000 MM MM  
CCCCCCCC 000000 MM MM

SYS

\$  
\$ REMOVE.COM  
\$

\$ This command procedure uses as input a data file (specified in P1 or  
\$ defaulted to SYSSUPDATE:VMSKITBLD.DAT on the system disk). The data file is  
\$ read as input commands to be executed by the procedure. The format of each  
\$ record is defined in VMSKITBLD.COM. P2 specifies the category of files to  
\$ be removed from the system disk. This is one of the options specified by  
\$ the UVMS qualifier in the data file. If P2 is not specified, then the user  
\$ is prompted for the option category.

\$ START:  
\$ ON ERROR THEN GOTO ALL\_DONE  
\$ ON CONTROL Y THEN GOTO ALL\_DONE  
\$ DEFDIR := 'F\$DIRECTORY()  
\$ DEFUIC := 'F\$USER()  
\$ DEFROOT := "SYSS\$SYSROOT:"  
\$ DEFEXE := "SYSS\$SYSROOT:[SYSEX]"  
\$ DEFUPD := "SYSS\$SYSROOT:[SYSUPD]"  
\$ WRITE SYSS\$OUTPUT ""  
\$ WRITE SYSS\$OUTPUT -  
\$ "This command procedure removes entire options or sub-options  
\$ WRITE SYSS\$OUTPUT -  
\$ "of the MicroVMS kit. Help may be obtained for any of the prompts"  
\$ WRITE SYSS\$OUTPUT "by typing HELP."  
\$ WRITE SYSS\$OUTPUT ""  
\$ WRITE SYSS\$OUTPUT "Valid options and sub-options are:"  
\$ WRITE SYSS\$OUTPUT ""

\$ Get data file name

\$ FILENAME = P1  
\$ IF FILENAME .EQS. "" THEN FILENAME = "SYSSUPDATE:VMSKITBLD.DAT"  
\$ FILE = FILENAME

\$ Find the valid values for OPTIONS and SUB-OPTIONS from the data file.

\$ KINDA\_VALID = "ALL,BASE,HELP,IGNORE"  
\$ VALID\_OPTION = ""  
\$ PREV\_VALID = ""  
\$ PREV\_SUB = ""  
\$ OPEN7READ DATAF 'FILENAME  
\$ ON ERROR THEN CLOSE DATAF  
\$ ON CONTROL\_Y THEN CLOSE\_DATAF  
\$ VALID\_REQUESTS:  
\$ READ/END OF FILE=NORMAL EXIT DATAF RECORD  
\$ IF F\$EXT(0,T,RECORD) .EQS. "\$" THEN GOTO END VALID  
\$ IF F\$EXT(1,1,RECORD) .NES. "\*" THEN GOTO VALID REQUESTS  
\$ VALID = F\$EXT(F\$LOC("!\* ",RECORD)+3,F\$LOC(" ",RECORD)-3,RECORD)  
\$ IF VALID .NES. PREV\_VALID THEN VALID\_OPTION = VALID\_OPTION + VALID + ","  
\$ IF VALID .NES. PREV\_VALID THEN PREV\_SUB = ""  
\$ IF VALID .NES. PREV\_VALID THEN WRITE SYSS\$OUTPUT " ",VALID  
\$ PREV\_VALID = VALID  
\$ VALID\_SUB = F\$EXT(F\$LOC(" ",RECORD) + 1,1,RECORD)  
\$ VALID\_VALID = PREV\_SUB + VALID\_SUB  
\$ PREV\_SUB = VALID\_VALID

```
$      WRITE SYSSOUTPUT "      ",RECORD
$      GOTO VALID_REQUESTS
$END_VALID:
$      CLOSE DATAF
$      ON ERROR THEN GOTO ALL_DONE
$      ON CONTROL_Y THEN GOTO ALL_DONE
$      VALID_DEL = ""
$      VALID_OPTION = 'VALID_OPTION + KINDA_VALID
$
$      OPTIONS = P2
$      SAVE_OPTIONS = OPTIONS
$      SUBOPTS = ""
$      SUB_OPTION = ""
$      SUB_OPT_RETURN := ALL_DONE
$      REMOVED_OPTIONS =
$      ENT_OPT = P3
$
$      | If option(s) not specified then get options list
$GET_OPTION:
$      IF OPTIONS .NES. "" THEN GOTO VALIDATE_OPTION
$      WRITE SYSSOUTPUT " "
$      IF REMOVED_OPTIONS .NES. "" -
$          THEN WRITE SYSSOUTPUT 'Option(s) previously removed: ',REMOVED_OPTIONS,''
$      WRITE SYSSOUTPUT " "
$      WRITE SYSSOUTPUT 'What layered VMS option(s) do you want to remove.'
$      INQUIRE OPTIONS 'Options must be separated by commas, e.g., USER,UTIL,SYSP'
$      SAVE_OPTIONS = OPTIONS
$      GOTO GET_OPTION
$
$      | Extract and validate an option, update options list
$VALIDATE_OPTION:
$      RETURN := GET_OPTION
$      OPTION = OPTIONS
$      IF F$LOC('' ,OPTIONS) .NE. F$LEN(OPTIONS) THEN OPTION = F$EXT(0,F$LOC('' ,OPTIONS),OPTIONS)
$EX_OPTION:
$      TEMP_1 = F$LOCATE(OPTION,VALID_OPTION)
$      IF (F$EXT(TEMP_1-1,1,VALID_OPTION) .NES. ".") THEN GOTO BAD_OPTION
$      IF TEMP_1 .EQ. F$LENGTH(VALID_OPTION) THEN GOTO BAD_OPTION
$      TEMP_2 = F$EXTRACT(TEMP_1,132,VALID_OPTION)
$      TEMP_2 = F$EXTRACT(0,F$LOCATE('' ,TEMP_2),TEMP_2)
$      IF OPTION .NES. TEMP_2 THEN GOTO BAD_OPTION
$      IF OPTION .EOS. "HELP" THEN GOTO HELP_OPTION
$      IF OPTION .EOS. "ALL" THEN GOTO BAD_OPTION
$      IF OPTION .EOS. "BASE" THEN GOTO BAD_OPTION
$      OPTIONS = OPTIONS - OPTION - ""
$      OPT_RETURN := ALL_DONE
$      IF OPTIONS .NES. "" THEN OPT_RETURN := GET_OPTION
$      TOTAL_OPT = OPTION
$      REMOVED_OPTIONS = REMOVED_OPTIONS + " " + OPTION
$      REMOVED_SUB_OPTIONS =
$
$      | Remove check - entire option or not
```

H 7  
UV

```
SENT_OPTION:  
  WRITE SYSSOUTPUT ...  
  IF ENT_OPT .EQS. "HELP" THEN GOTO HELP_ENT_OPT  
  IF ENT_OPT THEN GOTO CHECK_SEARCH  
  IF .NOT. ENT_OPT THEN GOTO GET_SUB_OPTION  
  INQUIRE ENT_OPT 'Do you want to remove the entire "OPTION" option (y/n)'  
  IF ENT_OPT .EQS. "" THEN GOTO ENT_OPTION  
  GOTO ENT_OPTION  
  
  | If not entire option then get sub-options list  
  
GET_SUB_OPTION:  
  RETURN := GET_SUB_OPTION  
  IF SUBOPTS .NES. "" THEN GOTO EX_SUB_OPT  
  WRITE SYSSOUTPUT ...  
  IF REMOVED_SUB_OPTIONS .NES. "" -  
    THEN WRITE SYSSOUTPUT "Sub-option(s) previously removed: ",REMOVED_SUB_OPTIONS,""  
  WRITE SYSSOUTPUT ...  
  WRITE SYSSOUTPUT "What sub-option(s) do you want removed for option ",OPTION,"."  
  INQUIRE SUBOPTS "Sub-options must be separated by commas, e.g., A,C,F"  
  SAVE SUBOPTS = SUBOPTS  
  IF SUBOPTS .EQS. "" THEN GOTO GET_SUB_OPTION  
  IF SUBOPTS .EQS. "HELP" THEN GOTO HELP_SUB_OPTION  
  
  | Extract and validate sub-option, update sub-options list  
  
EX_SUB_OPT:  
  IF SUBOPTS .EQS. "" THEN GOTO ALL_DONE  
  SUB_OPTION = SUBOPTS  
  IF F$LOC("",SUBOPTS) .NE. F$LEN(SUBOPTS) THEN SUB_OPTION = F$EXT(0,F$LOC("",SUBOPTS),SUBOPTS)  
VALIDATE_SUB_OPTIONS:  
  SUBOPTS = SUBOPTS - SUB_OPTION - ""  
  SUB_OPT_RETURN = OPT_RETURN  
  IF SUBOPTS .NES. "" THEN SUB_OPT_RETURN := GET_SUB_OPTION  
  VALID_OPT = VALID 'OPTION'  
  IF F$LOCATE(SUB_OPTION,VALID_OPT) .EQ. -  
    F$LENGTH(VALID_OPT) THEN GOTO BAD_SUB_OPTION  
  
  TOTAL_OPT = OPTION + " " + SUB_OPTION  
  REMOVED_SUB_OPTIONS = "REMOVED_SUB_OPTIONS" + " " + SUB_OPTION  
  
  | Is the SEARCH utility installed?  
  
CHECK_SEARCH:  
  SRCH = F$SEARCH("DEFEXE'SEARCH.EXE")  
  
  | If search for search command successful, then use search command  
  | on VMSKITBLD.DAT to create smaller file containing list of files  
  | to be deleted.  
  
SOPT_SRCH:  
  IF S .EQS. "" THEN GOTO OPEN_RECORD  
  SEAR 'FILENAME' 'UVMS=ALL','UVMS=""TOTAL_OPT-
```

```
S /OUTPUT='DEFUPD'UVOPTION.DAT
S FILE = ""'DEFUPD'UVOPTION.DAT"
S
S This is the main loop that is executed for each record in the data file.
S If search command used then FILE opened is UVOPTION.DAT. If search
S command not used then FILE opened is VMSKITBLD.DAT.
S
SOPEN_RECORD:
S OPEN/READ DATAF 'FILE
S ON ERROR THEN GOTO CLOSE_DATAF
S ON CONTROL_Y THEN GOTO CLOSE_DATAF
SRECORD_LOOP:
S READ/END OF FILE=NORMAL EXIT DATAF RECORD
S IF F$EXTRACT(0,1,RECORD) .EQS. "'S'" THEN GOTO RECORD_LOOP
S IF F$EXTRACT(0,1,RECORD) .EQS. """" THEN GOTO RECORD_LOOP
S IF F$EXTRACT(0,1,RECORD) .EQS. "",," THEN GOTO RECORD_LOOP
SGET_UVMS:
S UVMS = F$EXTRACT(F$LOCATE("UVMS=",RECORD)+5,132,RECORD)
S IF UVMS .NES. """" THEN GOTO PARSE_UVMS
S WRITE SYSSOUTPUT "%WARNING - No UVMS qualifier on record: "
S WRITE SYSSOUTPUT """""RECORD"""""
S GOTO RECORD_LOOP
S
SPARSE_UVMS:
S UVMS = F$EXTRACT(0,(F$LOCATE("",,UVMS)),UVMS)
S IF ENT OPT THEN TOTAL_OPT = UVMS
S SAVESET NAME = UVMS
S UVMS = F$EXTRACT(0,(F$LOCATE("",,UVMS)),UVMS)
S IF F$LOCATE(UVMS,VALID OPTION) -.NE. F$LENGTH(VALID_OPTION) THEN GOTO CHECK_OPTION
S WRITE SYSSOUTPUT "%WARNING - Invalid UVMS qualifier on record: "
S WRITE SYSSOUTPUT """""RECORD"""""
S GOTO RECORD_LOOP
S
S Check if this file is in the option we are deleting.
S
CHECK_OPTION:
S IF SAVESET_NAME .EQS. "ALL" THEN GOTO GET_NAME
S IF SAVESET_NAME .NES. TOTAL_OPT THEN GOTO RECORD_LOOP
S
S Get the name from the current record. (The name is always the first field
S in the record, and doesn't have a keyword associated with it.)
S
SGET_NAME:
S NAME = F$EXTRACT(0,F$LOCATE("",,RECORD),RECORD)
S
S Get the operation.
S
SGET_OPERATION:
S OPERATION = F$EXTRACT(F$LOCATE("OP=",RECORD)+3,132,RECORD)
S IF OPERATION .EQS. """" THEN GOTO NORMAL
S OPERATION = F$EXTRACT(0,F$LOCATE("",,OPERATION),OPERATION)
S GOTO 'OPERATION'
S
SSET:
S SOURCE_DEFAULT = "[" + ""NAME"" + "]"
SCREATE:
```

\*\*F  
\$ENTER:  
\$RENAME\_C:  
\$RENAME\_B:  
\$SYSGEN:  
\$ GOTO RECORD\_LOOP  
\$  
\$BUILD:  
\$COPY:  
\$NORMAL:  
\$CONTIG:  
\$ IF FSSEARCH("DEFROOT" "SOURCE\_DEFAULT" "NAME") .EQS. "" THEN GOTO RECORD\_LOOP  
\$ DELETE/LOG 'DEFROOT' 'SOURCE\_DEFAULT' 'NAME';\*  
\$ IF NAME .EQS. "STARLET.OLB" THEN LIBR/CREATE 'DEFROOT' 'SOURCE\_DEFAULT' 'NAME'  
\$ GOTO RECORD\_LOOP  
\$  
\$NO\_DATAF:  
\$ WRITE SYSSOUTPUT " Unable to open data parameter file."  
\$ GOTO ALL\_DONE  
\$  
\$HELP\_ENT\_OPT:  
\$ WRITE SYSSOUTPUT ""  
\$ WRITE SYSSOUTPUT "Typing Y indicates you want the entire option"  
\$ WRITE SYSSOUTPUT "to be removed. Typing N indicates you want only"  
\$ WRITE SYSSOUTPUT "portions of the option to be removed. The prompt"  
\$ WRITE SYSSOUTPUT "for these sub\_options follows if you type N."  
\$ WRITE SYSSOUTPUT ""  
\$ ENT\_OPT = ""  
\$ GOTO ENT\_OPTION  
\$BAD\_OPTION:  
\$ WRITE SYSSOUTPUT ""  
\$ WRITE SYSSOUTPUT "You specified an invalid option in: ",SAVE\_OPTIONS,""  
\$ WRITE SYSSOUTPUT ""  
\$HELP\_OPTION:  
\$ WRITE SYSSOUTPUT "The valid options are:  
\$ WRITE SYSSOUTPUT ""  
\$ VALID = VALID\_OPTION - KINDA\_VALID  
\$BACK:  
\$ TEMP = FSEXT(0,FSLOC("",",VALID),VALID) ",TEMP  
\$ VALID = VALID - TEMP - ""  
\$ IF VALID .NES. "" THEN GOTO BACK  
\$ OPTIONS = ""  
\$ WRITE SYSSOUTPUT ""  
\$ WRITE SYSSOUTPUT "Options must be separated by commas."  
\$ WRITE SYSSOUTPUT ""  
\$ GOTO 'RETURN'  
\$BAD\_SUB\_OPTION:  
\$ WRITE SYSSOUTPUT ""  
\$ WRITE SYSSOUTPUT "You specified an invalid sub-option in: ",SAVE\_SUBOPTS,""  
\$HELP\_SUB\_OPTION:  
\$ WRITE SYSSOUTPUT ""  
\$ WRITE SYSSOUTPUT "The valid sub-options for ",OPTION," are:"  
\$ WRITE SYSSOUTPUT ""  
\$ IF SRCH .NES. "" THEN GOTO SRCH\_SUB\_OPT\_HELP  
\$ OPEN/READ DATAF 'FILENAME'  
\$ ON ERROR THEN GOTO CLOSE\_DATAF

```
S      ON CONTROL_Y THEN GOTO CLOSE_DATAF
$HELP_READ:
S      READ/END OF FILE=HELP FINISH DATAF RECORD
S      IF FSEXTRACT(0,1,RECORD) .NES. "!" THEN GOTO HELP_FINISH
S      IF FSEXTRACT(1,1,RECORD) .NES. "*" THEN GOTO HELP_READ
S      VALID = FSEXT(F$LOC("!* ",RECORD)+3,F$LOC("!",RECORD)-3,RECORD)
S      IF VALID .EQS. OPTION THEN WRITE SYSSOUTPUT "",RECORD
S      GOTO HELP_READ
$HELP_FINISH:
S      CLOSE DATAF
S      ON ERROR THEN GOTO ALL_DONE
S      ON CONTROL_Y THEN GOTO ALL_DONE
S      GOTO HELP_CONT
$SRCH_SUB_OPT_HELP:
S      STRING = "**** + "!* " + OPTION + "_" + ****"
S      SEARCH 'FILENAME' STRING
$HELP_CONT:
S      WRITE SYSSOUTPUT ""
S      WRITE SYSSOUTPUT "Sub-options must be separated by commas."
S      SUBOPTS := ""
S      GOTO 'RETURN'
$NORMAL_EXIT:
S      CLOSE DATAF
S      ON ERROR THEN GOTO ALL_DONE
S      ON CONTROL_Y THEN GOTO ALL_DONE
S      IF TOTAL_OPT .NES. (OPTION + "_" + SUB_OPTION) -
          THEN TOTAL_OPT = OPTION
S      WRITE SYSSOUTPUT ""
S      WRITE SYSSOUTPUT "***** ",TOTAL_OPT," REMOVED *****"
S      WRITE SYSSOUTPUT ""
S      GOTO 'SUB_OPT_RETURN'
$CLOSE_DATAF:
S      CLOSE DATAF
$ALL_DONE:
S      IF FSSEARCH("DEFUPD'UVOPTION.DAT") .NES. "" THEN DELETE 'DEFUPD'UVOPTION.DAT;*
S      SET DEFAULT 'DEFDIR'
S      EXIT
```

MOD  
SA  
P  
SD  
DUM  
VMS  
VMS  
KFI  
GBL  
GBL  
MAX  
SYS  
IF  
IF  
BAL  
IRP  
IRP  
WSP  
NPA  
NPA  
PAG  
SPT  
LRP  
LRP  
SRP  
SRP  
MPW  
MPW  
SWP  
FRE  
FRE  
GRO  
BOR  
CLI  
LOC  
LOC  
RES  
DEA  
SCS  
SCS  
SCS  
PAM  
PAP  
CJF  
TTY  
RMS  
ACP  
ACP  
ACP  
ACP

0431 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

